A System for the Functional Testing and Simulation of Custom and Semicustom VLSI Chips

E. M. Olson and L. J. Deutsch
Communications Systems Research Section

This article describes a system for the functional testing and simulation of custom and semicustom very large scale integrated (VLSI) chips that are designed using the integrated UNIX-based computer-aided design (CAD) system. The testing and simulation system consists of two parts. One of these is a special purpose hardware device that is capable of controlling the digital inputs and outputs on a custom chip. This device, the Digital Microcircuit Functionality Tester (DMFT) system, can be operated by itself or in conjunction with the VAX host computer on the CAD system. The DMFT is intetegrated into a microprobe station so that these signals can be injected or read from nodes inside the chip, as well as at the pins. The second part of the system is a software package that is installed on the VAX. This software package, "logic," includes a full-screen editor for developing chip test sequences and drivers for both the DMFT and the "esim" logic simulator.

I. Introduction

The design and testing of custom and semicustom VLSI chips for the Deep Space Network (DSN) Advanced Systems Program at JPL is accomplished on an integrated UNIX-based CAD system. This system, as described in [1], comprises various software tools and hardware peripheral devices that facilitate the design, simulation, verification, and testing of very complex microcircuits in a number of different solid-state technologies. This article is devoted to the portion of the system that is used to perform functional testing of the microcircuits after their fabrication.

One of the most difficult tasks in the production of custom or semicustom digital microcircuits is verifying that the completed part performs the function for which it was designed. This is, in general, a long and laborious process that involves the use of many varied pieces of test equipment. The following is a description of how this process was performed before the present system was completed.

Chips would first be examined under a microscope to weed out those that were obviously bad. After this process, the remaining chips would be tested for functionality. In order to prove that a chip is performing the desired function, it is often necessary to submit the chip to many thousands of bits of data while monitoring the chip's outputs. This has been done in the past by using digital word generators for input and oscilloscopes or logic analyzers for output. Even the best word gen-

erators, however, are limited in the number of bits they can store. A typical generator can produce signals that are up to 1024 bits in length. This number is inadequate for most of the chips that are being designed in this effort. A Reed-Solomon encoder chip [2], for example, requires at least 2500 bits of non-periodic data for even the simplest test. An alternative is to build special hardware "jigs" for each chip that produce the desired bit sequences. This takes a lot of time and can be very expensive. In either case, there is no good way for the user to define the sequences that are needed. Finally, the output signals, as seen on the oscilloscope or logic analyzer, are not easily stored for future use. These problems have been essentially alleviated by the development of the Digital Microcircuit Functionality Tester (DMFT) and the "logic" software package.

The DMFT has been developed as part of the JPL Advanced Systems Program specifically for incorporation into the integrated UNIX-based CAD system. The tester was built internally to meet specific budget demands that weren't addressable by testers of similar functionality and speed that were currently on the market. Furthermore, since the cost of developing the tester was relatively small, the VLSI design engineers in this program were afforded a chance to develop requirements for future testers from first-hand experience and evaluation of this prototype tester. Since the tester was inexpensive, the cost of developing this first-hand experience is considerably less than it would cost to rent or purchase a tester from the current market. The DMFT has also allowed these design engineers to begin developing standards and requirements for testers to be incorporated into the system in the future, while at the same time addressing the pressing need for test capability at the present.

The DMFT consists of special purpose hardware and firmware for the generation and monitoring of digital signals to and from a VLSI chip. It is capable of generating eight independent sequences of up to 4096 bits at a time. The DMFT also generates several clock signals that are synchronized to these data. It can subject these to the chip being tested (sometimes called the device under test or DUT) at selectable clock rates of up to 20 MHz. This maximum frequency is sufficient at present because all the design work that is currently going on involves MOSFET [3] devices and fabrication techniques that cannot operate any faster than this. The DMFT can monitor eight chip outputs for up to 4096 bits at a time. The firmware in the DMFT supports a terminal interface so that the user may define and edit the test vectors, run tests, and display the output. In addition, the firmware includes the support of a host computer interface with asynchronous protocol so that the DMFT may be completely controlled by the VAX computer.

This special purpose hardware and firmware is built into a specially designed microprobe station. The station consists of a high-power stereoscopic microscope for examining the device under test, a set of four probes that can be positioned on points inside the actual chip, and an image-shearing device for measuring the thickness of layers on the chip. The DMFT is mounted so that the chip being tested is positioned under the microscope. Signals to and from the chip can be routed through the chip's pins and/or through any of the probes. This means that the DMFT system may be used for interactive debugging of bad designs by tracing signals back through the chip to find problem areas.

The logic software package runs on the UNIX-based CAD system. It was written here at JPL so that users of the CAD system could easily control the DMFT from any remote site. The package is fairly comprehensive and includes a full-screen editor for defining, displaying, and modifying large sets of test vectors. The format of the editor is similar to that of many commercial logic analyzers and is, therefore, easy for the designers to use. The sequences may be stored on a disk and read back later. In this way, the designer can build a large database of test vectors for a design. The logic program can use these vectors to run the DMFT and display the results in real time. The results can also be stored on the disk or printed on a line printer. Logic can also use the vectors to simulate a chip before fabrication. This is done by automatically invoking the "esim" logic simulator [1, 4]. By using this capability, a designer does not have to learn two separate systems for simulation and testing. Also, the designer can use the same test vectors for testing as were used in simulation. The results can be compared automatically using the UNIX compare utilities to verify the functionality of the finished chip.

The remainder of this report describes the tester system in detail. Section II describes the testing philosophy behind the design. Section III contains an overview of the DMFT hardware. Sections IV and V describe the DMFT hardware in detail. Section VI describes the logic software package. Section VII is devoted to an assessment of the current system and plans for future generation testing systems.

II. The Testing Philosophy

Integrated circuit testing is a very broad field and incorporates many disciplines and technologies. In order to fully verify that an integrated circuit is qualified for its application, it must not only pass electrical specifications, but also mechanical and cosmetic specifications. Currently the Advanced Systems Program VLSI design effort is directed to research and development of prototype LSI and VLSI architectures and their applications. As such, this effort is currently not con-

cerned with mechanical testing, and is concerned with cosmetic testing only for the purpose of rejecting obviously bad dies. Also, the output of effort to date has exclusively been digital MOSFET designs. Thus, the testing effort is further limited primarily to digital qualification at speeds under 20 MHz. Although parametric testing (i.e., testing that considers the analog performance of devices) is a very important step in the qualifying of digital circuits, the tester described herein does not have any capabilities to directly measure currents, voltages, or any other analog qualities of an integrated circuit – it has been designed only to test for logical functionality.

Logical testing of digital integrated circuits can be oversimply characterized as the systematic presentation of stimulus vectors to the device under test, while systematically recording response vectors from the device. Vectors consist of sets of bits either presented to (stimulus) or read from (response) the device. The voltage-to-logic-level correspondence is preset to some range for all the bits in a vector (for example 0 to 0.8 volts on a pin for a logical "0" and 2.4 to 5.0 volts for a logical "1"). Usually each bit in a test vector corresponds physically to the logic level that is presented to (or recorded from) a particular pin on the device. The vector itself can be thought of as a copy of the logical status of some particular pins on the device at some point in time or during an interval of time.

III. The DMFT – An Overview

The DMFT hardware and firmware are housed in an aluminum chassis that mounts under a microprobe station platform (see Figure 1). The entire electronics of the tester is contained in this chassis, including three power supplies, front panel control logic, and a standard multibus (IEEE standard number 796) circuit board on which most of the circuitry itself is mounted. A 64-pin zero insertion force socket and a signal patch panel are also mounted on the multibus board. The socket and the patch panel protrude through an opening in the probe platform allowing access to the device being tested with the microscope and probing equipment, while maintaining the bulk of the tester's electronics out of sight. The test setup is controlled partly by front panel controls and partly by setup through the tester firmware via either a host computer or a standard RS-232 (CCITT recommended standard number V.24) terminal (this later mode of operation is called stand-alone).

Figure 2 shows the layout of the front panel controls and indicators. Their functions are described as follows:

Power-On Indicator – This indicator is lit when the DMFT system tester power is on

Slow Clock Indicator – This indicator is lit when the slow internal clock range is selected; if it is not lit, then the fast internal clock is selected

Test Indicator – This indicator is lit while a test is actually in progress

Internal/External Clock Switch – This switch is used to select either the internal DMFT clock or an external clock for testing

External Clock In – This is a connector for external clock input

Frequency Adjust – This is a Vernier dial that is used to set the internal clock frequency

Clock Out – This is a connector that provides the selected clock signal as an output

DUT Voltage Adjust – This is a Vernier dial that is used to adjust the power supply voltage from 1.5 to 7 volts for the device under test

DUT Current Meter – This is a meter that monitors the electrical current being drawn by the device under test

DUT Voltage Meter – This is a meter that monitors the voltage applied to the device under test

Main Power Switch – This is the main power switch for the DMFT

DUT Power Switch – This switch is used to turn power, to the device under test, on and off

Reset Button – This push-button resets the entire tester and restarts tester firmware.

A block diagram of the DMFT is shown in Fig. 3. It is clear from the figure that the concept is fairly simple. As mentioned above, the entire tester electronics including a 64-pin zero insertion force socket and a signal patch panel, fit on one multibus board. The tester hardware itself can be divided into two sections. The first section consists of the MUX, DEMUX, Vector Memory, CNTR ADDR GEN, and CLOCK GENERATOR sections of the block diagram and will be called the "Vector Buffer Section." The second section of the tester consists of the controlling microprocessor and its associated support circuitry, and also includes the firmware contained in the microprocessor's memory. It will be called the "Control Section." These two portions will be discussed in the following two sections.

An abbreviated list of the important DMFT hardware specifications appears in Table 1.

IV. The Vector Buffer Section

The Vector Buffer Section (VBS) of the tester is really the core of the tester and is that part of the hardware that actually presents and records stimulus and response vectors during a test. The tester's VBS and the test itself are entirely synchronized to the Clock Generator's master clock. The Vector Memory is designed to be able to transmit and receive vectors at a rate of 20 MHz (one vector output and one received every 50 nS). The memory is actually implemented on 4K-deep-by-8-bits-wide, 45 nS, static random access memory (SRAM) chips and are arranged as 4 groups of 4K-deep-by-8-bits-wide chips. The idea behind the organization and support circuitry is to maintain two groups of two parallel memories, one group to drive vectors to the device under test and one group to receive vectors from it. The two parallel memories within each group are cycled 180 degrees apart, allowing the DMFT to receive and return vectors at twice the rate of which one individual memory is capable. This organization and clocking scheme is effected in the following manner.

Memories C and D each contain test vectors to be sent to the device being tested. These vectors are placed in the two memories in such a way that memory C contains the data for odd-numbered time slots and memory D contains the data for even-numbered time slots. The Control Section handles this ordering such that it is transparent to the user. Memories A and B each contain vectors returned from the device being tested as the results of prior test vectors. Those resultant vectors are also ordered in memories A and B the same way the odering takes place in C and D. Again, upon reading the resultant vectors the multiplexed organization is transparent to the user by manipulation through the Control Section. Stimulus vectors from buffers C and D are synchronously multiplexed to the device under test while response vectors are synchronously multiplexed to buffers A and B.

Control of the addressing of the Vector Memories is effected by the Address Generators A and B. Initial addresses for the memories are loaded into presettable counters by the Control Section, commanded by the user. The initial address determines the starting point for a test or a vector transfer from the Vector Memories to the Control Section and on to the outside world. The ending address for a test is also loaded into the Counter Address Generator block by the Control Section, and the loaded value determines the length of a test. When the ending address has been reached, the end of test interrupt is asserted and the VBS is returned to the Control Section.

All clocks in the vector buffer section are derived in some way from the output of the internal voltage-controlled oscillator (VCO) or the external clock. The internal VCO is selectable for a slow or fast range through the Control Section. Since everything in the VBS is synchronized to the master clock, the use of the internal VCO or an external clock, as the master, allows continuously variable test rates of up to 20 MHz. The clock rate for the internal VCO is a few kHz to about 20 MHz. Additionally, since the entire VBS is treated as an asynchronous input/output device by the Control Section, the tests run completely asynchronously to the Control Section and the microprocessor is not required to do real-time work for the performance of tests. The actual frequency of the internal clock can be set by the front panel Frequency Adjust and monitored through the Clock Out connector.

V. The Control Section

The Control Section consists of an Intel-Z80A 4-MHz microprocessor, a serial input/output controller (SIO), a counter timer circuit (CTC), some address decoding logic, some interrupt arbitration logic, some buffers and latches, and the Z80's core memory. The CTC is programmed to interrupt the microprocessor every second for user timing features. The CTC also provides baud rate clocking to the SIO for both SIO ports. Interrupt arbitration is handled internally by the SIO (highest priority) and CTC (second priority) and externally for the End-Of-Test (EOT) signal by the interrupt control logic. The SIO and CTC are programmed to present their own interrupt vectors with their interrupts. The EOT interrupt vector is presented at the appropriate time in conjunction with the end of a test.

The primary function of the Control Section is to act as the interface between the Vector Buffer Section, which actually performs tests, and the user who desires a test. The user can be either the host computer or someone at a terminal connected directly to the tester (stand-alone mode). The use of the microprocessor in the Control Section, along with the SIO controller allows a convenient interface to the tester through an RS-232 standard interface. The Control Section hardware and firmware combine to allow the user to generate and input test vectors, and to view the stimulus and response vectors after a test, in any one of three formats in the stand-alone mode. Also in the stand-alone mode, the user has complete control over the tester through a simple monitor program that, among other things, allows the user to enter and execute his own firmware for special configurations or testing. The Control Section hardware and firmware also handles all high-speed data transfers and handshaking between the host computer and the tester. Finally, the Control Section handles all the interpretation of commands for test setup including the time interlacing of vectors as mentioned above in the vector buffer discussion.

In addition to the visible functions the Control Section performs, there are several other functions that are handled by the Control Section that are not as readily apparent. For example, on power up or reset, the vector memories are tested and bad memory chips are called out by part number, permitting quick diagnosis and replacement of bad parts. The Control Section also keeps track of the length and duration of tests and insures that commands aren't accidentally executed during a test run that might interfere with the test results. The Control Section buffers all commands internally allowing the "repeat-last-command" function and correction on the fly. Special commands available to the user through the standalone terminal permit non-periodic and variable length periodic stimulus vectors to be generated quickly with a single command. The tester is capable of detecting and flagging erroneous data transfers or requests from the host computer, adding further confidence to test results. The host computer interface is defined entirely in the American Standard Code for Information Transmission (ASCII), allowing the tester to be used with virtually any host computer that can implement the command protocol over an RS-232 interface.

VI. The Logic Software Package

The logic software package was written to run on the UNIX-based CAD system and to allow the user to create, edit, store, and retrieve test vectors and to run both the esim logic simulator and the DMFT. The program is interactive and it may be run from any user terminal on the CAD system. This means that both simulation and testing of a chip design may be performed from the user's office. The user must be communicating to the host VAX computer on an ANSI standard (American National Standards Institute documents X3.4 and X3.64) terminal. All of the terminals that are currently produced by the Digital Equipment Corporation (such as the VT100 and VT200 series terminals) meet this standard as well as most personal computer terminal emulation programs. In fact, logic has been run on many different terminals including Apple and IBM personal computers. A photograph of the logic program screen in use appears in Fig. 4.

The logic program has three modes of operation. These are called the "display," "edit," and "test" modes and they are each described below. Each of these modes has a set of single letter commands associated with it. The appropriate command menu for the current mode is always displayed in the upper portion of the terminal screen. Most of the remainder of the screen is used to display a 16-line by 64-bit window on the test vectors. The program has the capacity to handle 80 lines of 4096 bits – and this can be easily changed by recompiling the code. The bottom line of the screen is the "command" line. It is used by the user to enter various parameters during the operation of the program and to display error messages and warnings. The very top part of the display contains various pieces of useful information including the name of the file being edited, the current bit, the number of inputs and outputs in the current set of test vectors, and the current mode.

The display mode comprises a mechanism for scanning through the test vectors that are currently loaded into memory. It also contains commands for writing vectors to the host disk and reading vectors from the disk and for obtaining a hardcopy printout of the current vectors (also in logic analyzer format). The screen contains a cursor that points to the current signal (line) and bit. The commands in the display mode are actually available to the user in all three modes. The display menu only appears in the display mode.

The second mode in the logic program is the "edit" mode. The edit mode contains a set of commands that are used to create test vectors and to modify them. Each signal, or line, in the display can be made either an "input" or "output" signal. Input signals are stimuli to the simulator and tester while output signals are responses. Figure 4 shows the edit mode screen. Commands are included for a comprehensive set of editing functions including generating arbitrary periodic sequences, moving and copying signals, and even producing random signals.

The third and final mode in the logic program, the "test" mode, is used to run the simulator and the tester. When the user enters the test mode, a consistency check is run on the current vector database to insure that the vectors represent a valid test. Among the things that are checked are multiple signals with the same name, signals with undefined bits in them (i.e., the user failed to enter a one or zero at some location), and that the signals are all of the same length. In each of these cases, logic lets the user decide if the problem should be fixed automatically. For example, if a particular signal has a greater bit length than the rest, logic can truncate that signal if desired. Once the consistency check has completed without errors, the test mode is brought into operation. Either the esim simulator or the DMFT may be run with the existing data from this mode. Additional data, however, are required to run either of these and the test mode offers the user the ability to merge the necessary data at this point.

If the user requests a run of the esim simulator, the logic program will ask the user for the name of a file set that contains the netlist information of the design to be simulated. Logic then checks this file set to be certain that all the signal names that have been defined in the current test vector set actually appear in the design. If they do not, this information is made available to the user. If they are, then esim is run and the results are displayed on the screen. Esim output consists of 1's, 0's, and X's. The X's represent unresolved states. Logic

will display the X's and allow them to be saved, read, or edited just as the other logical states.

If the user wants to run the DMFT, then information is required that tells the logic program which signals are wired to each channel (of the eight input and eight output channels) on the DMFT. This information can be created by invoking the "edit pin file" command or by reading it in from an existing file. Before the DMFT is actually run, a consistency check on the pin list and the signal set is performed to make sure that no more than eight inputs and eight outputs have been requested and that inputs are wired to input channels and outputs to output channels. Assuming that all is fine, then the tester is run. The results appear on the screen just as in the case of esim, with the exception that X's cannot occur.

If the user wants to run another test (either with esim or the tester), the outputs can be reset in the test mode, also.

VII. Conclusions and Future Work

The system for testing and simulation that has been described in this report has been operational since February of 1985. In the short time since the system has been in use, it has proven to be of essential importance to the microcircuit projects in the Advanced Systems Program. It was used to test the Reed-Solomon encoder chips in only two weeks time. These chips included two test chips and three versions of the complete system chip. The chips all worked as expected – but this would have been next to impossible to determine without the DMFT. One other chip, part of a Hopfield neural memory system [5] being developed on a Director's Discretionary Fund grant, was designed, simulated using the logic program, and tested using the DMFT and logic, bringing to fruition the design and test methodology described in this report. The design and simulation of the chip took only two weeks and the testing was completed in just two days.

Some limitations of the system have already been discovered. Many engineers had suggestions for certain commands to be added to the logic program. These included the random signal generator and the undo command (which undoes the last edit operation). These have already been added to the program. Also, the need for external clock input protection circuitry for the DMFT was discovered and this too was added.

Among the development activities planned for the system are expanding the DMFT's capability in a second-generation device that will have many more input and output channels and automatic crossbar switching to replace the patch panel. Also, the logic program could be expanded to allow other simulators (such as "rsim" [6] or JPL in-house programs such as "ULYSES") to take advantage of the user-friendly interface.

The DMFT hardware is well documented. The documentation includes complete logic diagrams, a hardware manual, and a user manual. The DMFT can be duplicated for about $1400 including JPL engineering costs. The design is available to other parties at JPL who may wish to use it. The logic software package is written entirely in the "C" programming language, but it relies heavily on the UNIX operating system and the ANSI terminal standard. Those functions that are operating-system dependent are separated into a single C source module just as those for ANSI standard are. This means that the program might be rewritten to run on either a different operating system or terminal type by changing these files only. Logic, like the tester, is available to other parties at JPL.

Acknowledgments

The authors wish to acknowledge the help of C. R. Lahmeyer and J. J. O'Konek of the Communications Systems Research Section for their help in setting up and debugging the DMFT hardware and for some improvements on its design. Without their assistance, this job would have been much more difficult.

References

1. Deutsch, L. J., "An Integrated UNIX-Based CAD System for the Design and Testing of Custom VLSI Chips," *TDA Progress Report 42-81*, Jet Propulsion Laboratory, Pasadena, California, May 15, 1985, pp. 51-62.

2. Truong, T. K., Deutsch, L. J., Reed, I. S., Hsu, I. S., Wang, K., and Yeh, C. S., "The VLSI Design of a Reed-Solomon Encoder Using Berlekamp's Bit Serial Algorithm," Proceedings of the Third Caltech Conference on VLSI, California Institute of Technology, Pasadena, California, 1983, pp. 303-330.

3. Mead, C. and Conway, L., *Introduction to VLSI Systems*, Addison-Wesley Publishing Company, Menlo Park, California, 1980, pp. 1-37.

4. Baker, C. and Terman, C., "Tools for Verifying Integrated Circuit Designs," *Lambda*, Fourth Quarter, 1980.

5. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proceedings of the National Academy of Sciences, Vol. 79, April 1982, pp. 2554-2558.

6. Terman, C. J., "RSIM – A Logic-Level Timing Simulator," Proceedings of the IEEE Conference on Computer Design: VLSI in Computers, IEEE Computer Society Press, Silver Spring, Maryland, 1983.

Table 1. DMFT Specifications

Power requirements	120 VAC rms at 60 Hz and 2 Amps
Terminal connection	RS-232 standard, 9600 baud
Host computer connection	RS-232 standard, 9600 baud
Stimulus vector	8 user-definable logic bits wide, TTL logic levels
Response vector	8 bits wide, TTL logic levels
Power to device under test	1.5 to 7 Vdc at up to 100 mAmps
Minimum time between stimulus vectors or response vectors	50 nanoseconds
Maximum time between vectors	Infinite (dc)
Resolution of vector separation time	Infinite, continuously variable
Maximum number of vectors per test	8190 (4095 stimulus, 4095 response)
Minimum number of vectors per test	4 (2 stimulus, 2 response)

Fig. 1. The Digital Microcircuit Functionality Tester (DMFT) and microprobe station

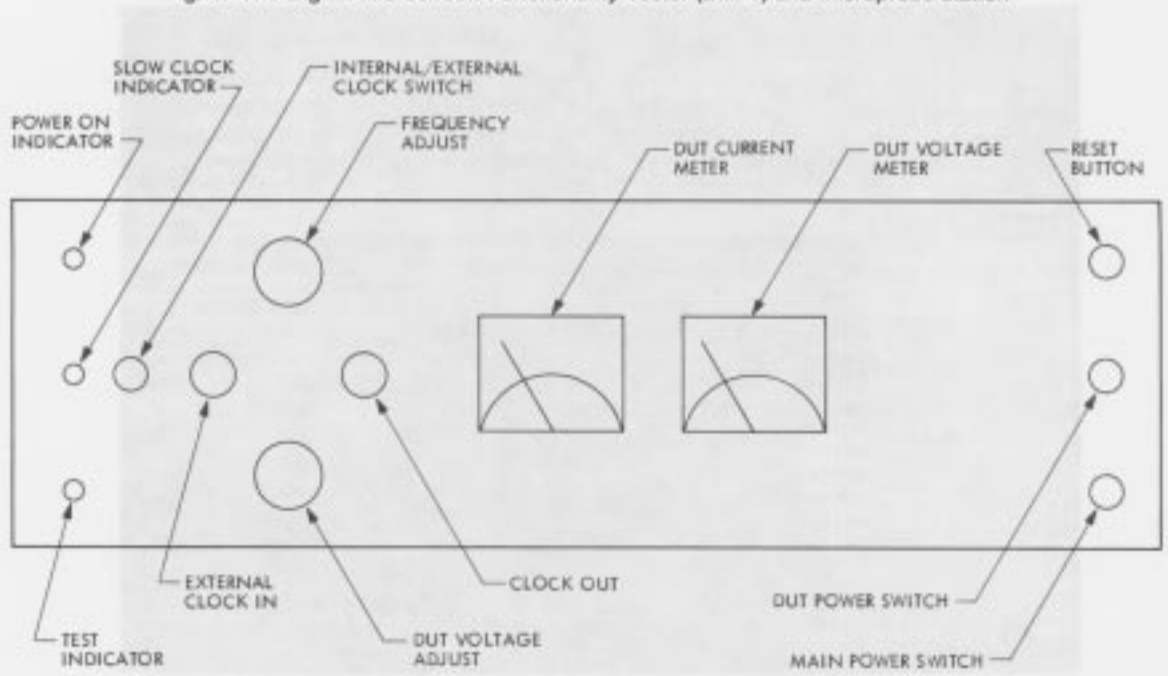

Fig. 2. Front panel controls and indicators on the DMFT

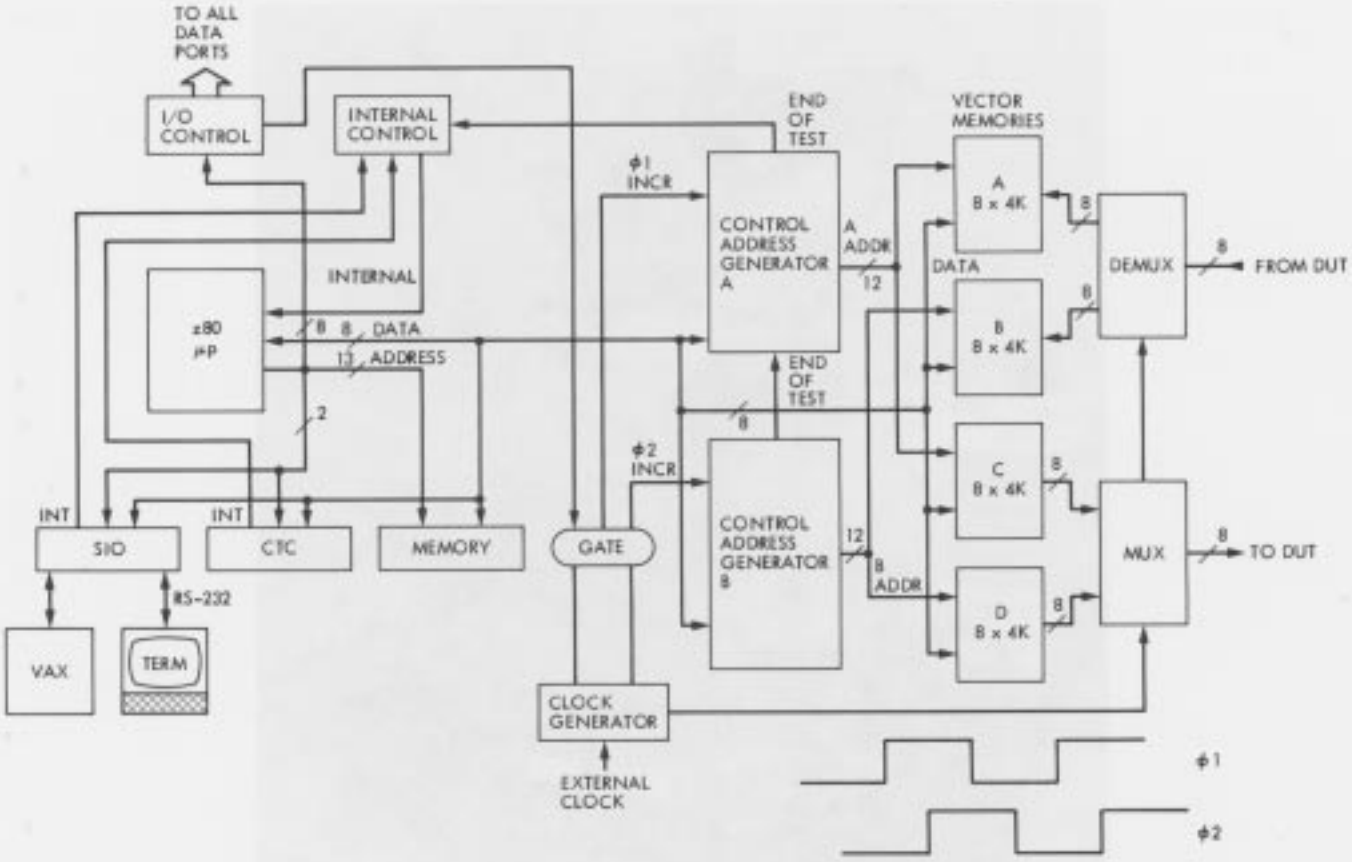

Fig. 3. System block diagram for the DMFT

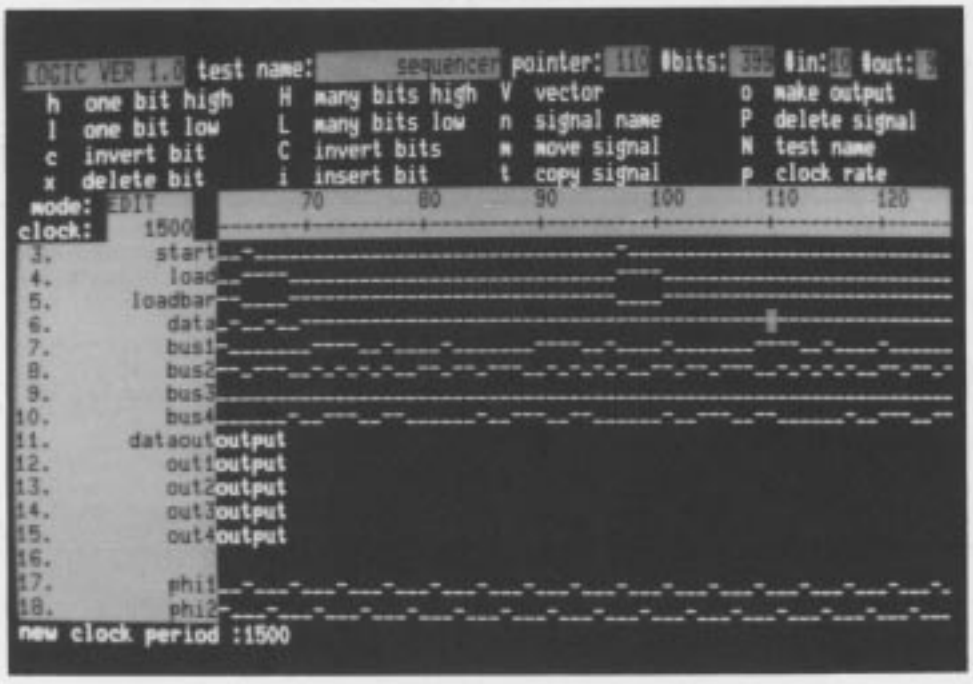

Fig. 4. The logic program display screen